

## Rambus Invention Disclosure Form

## Inventorship

JUL 9 2006

Inventor Name: Ken Chang ext.5322Residence Address: 396 Ramon Dr, Los Altos, CA 94024Citizenship: Taiwan (ROC) Division: (e.g., MATD, PATD, CCPD, ...) NCDInventor Name: Kevin Donnelly ext.5223

Residence Address:

Citizenship: USA Division: NCD

Inventor Name: ext.

Residence Address:

Citizenship: Division:

Inventor Name: ext.

Residence Address:

Citizenship: Division:

(If you believe that there are more than four inventors, please contact the IP group)

## Potential Filing Deadlines

Invention in or to be incorporated in any Rambus product designs? YesName of product design: Rambus SerDes Cell (RaSer™ Cell)Date product design released or scheduled for release? Not known yet, JNI contract pendingInvention disclosed or scheduled to be disclosed outside Rambus? NoNature of disclosure (e.g., customer meeting, white paper)? Customer meeting, data sheet

Date of disclosure or scheduled disclosure?

## Invention Disclosure

Title of Invention: A Method for Supporting both DDR and SDR Transceivers

Use the following outline to describe your invention, preferably within two or three pages. Include/attach any diagrams or other documents helpful to understanding the invention.

## 1. Problem addressed by the invention

To support dual rate or wide range operations for SerDes. The transmitter and receiver can run at a different rate on the same SerDes. This key in the invention is to solve the problem without major change in the current SerDes design.

2. Description of the invention and how it overcomes/mitigates the problem  
Please see attachment

## 3. Advantages of the invention over prior techniques

Please see attachment

4. Value of the invention to Rambus  
Please see attachment

*Records of Invention*

Serial nos. and page nos. of notebooks or other records containing notes and date of invention: \_\_\_\_\_  
In the schematic of rsc3/4/5/6/7/8, csc1, csc2.

Are there any non-documentary items useful to establish conception of the invention or reduction of the invention to practice? (e.g., models, prototypes, etc.) \_\_\_\_\_ If yes, please explain: \_\_\_\_\_

**ATTACH COPIES OF ANY DOCUMENTS USEFUL TO ESTABLISH DATE OF INVENTION (e.g., invention notebook pages, email prints, test/simulation results, draft specifications, etc.)**

Name of individual submitting disclosure (please print): Ken Chang \_\_\_\_\_

Signature: \_\_\_\_\_ Date: 8/24/2001 \_\_\_\_\_

Witness Signature: \_\_\_\_\_ Date: \_\_\_\_\_

## A METHOD FOR SUPPORTING BOTH DDR AND SDR TRANSCEIVERS

Ken Chang and Kevin Donnelly

### 1. Introduction

In high-speed transceivers, to avoid high clock rate, one commonly applies a DDR (double-data-rate) technique. In this case, the data runs at twice the speed of the clock. That is, the data is transmitted and received on both edges of the clock. Hence, the PLL or any clock generation circuit only needs to run at the half rate of the data speed.

In some applications, the transceivers have to be able to operate over a wide range, for instance, from 500Mb/s to 3.5Gb/s. With the DDR transceiver, the PLL or clock circuits have to run from 250Mb/s to 1.7Gb/s. While it is possible to design a circuit with such a wide range, the circuit performance, such as clock jitter, is usually compromised and thus is inferior to one designed for a narrower range. One experimental favorable range for a PLL is about 1:3. In the example, one may have to use two PLLs with one covering high data rates from 1.2Gb/s - 3.5Gb/s and one for low data rates from 500Mb/s to 1.5Gb/s.

In some other applications, the same PLL circuit has to support both ranges of data rates. One example is a Fiber Channel transceiver, where the transmitter and receivers can run at any combination of 2Gb/s and 1Gb/s simultaneously. Since the link is duplex, including both transmitter and receiver, using one PLL circuit for both the transmitter and the receiver can save power, area, and design effort. However, the PLL can only run at one rate but has to support both 2Gb/s and 1Gb/s data rate.

This invention is motivated by the two requirements addressed in the last two paragraphs. It allows the use of one PLL without the need for a wide range of operation. Above all, it has minimum impact on the jitter of the high-speed clocks.

## 2. Operation/Embodiments

The basic idea of this invention is that one can operate the transceivers at both DDR and SDR (Single-Data-Rate) mode. In the FiberChannel example, the VCO of the PLL runs at 1GHz. The transceiver is in DDR mode at 2Gb/s mode, and is in SDR mode at 1Gb/s. *Therefore, the PLL only needs to be optimized for one range: the highest possible needed. The transition between DDR to SDR is performed by digital logic (mainly multiplexors for data) which is added after the deserializer and serializer in the low speed domain. There is no extra loading or circuit added in the high-speed clock path. Therefore, the high-speed clock jitter is not increased.*

The following embodiment is based on the following design assumption. But it should not be limited to such design.

A dual-loop PLL with digitally adjusted phase mixers is used for the clock generator. A phase control logic is used to adjust the phase mixers for the CDR (Clock Data Recovery) clock. The transmit clock is directly tapped out from the VCO buffers. A divide-by-10 ratio is used between the reference clock and the VCO clock. The 10-to-1 serializer and 1-to-10 deserializer are used.

This invention is adding circuits without changing any of the above circuits, including PLL, CDR, transmitter, serializer and deserializer.

The discussion of the embodiments are separated into two independent sessions: CDR and transmitter.

### 2.1 CDR embodiment

The CDR discussion is further based on (but not limited to) the following:

Four receivers are used for the DDR mode: two for the data and two for the timing information. The clock for the data receivers is centered at the middle of the data eye while the clock for the timing receivers is centered at the edge of the data eye. Each deserializer is composed of even and odd paths. There are two deserializers, one for data and one for timing. Overall, there are four data paths: even data, odd data, even timing, and

odd timing. To further simplify the discussion, the data rates of interest are 2Gb/s and 1Gb/s.

Figure 1 shows the clocking for both 2Gb/s and 1Gb/s when CDR is locked.  $dclk$  and  $\overline{dclk}$  represent the data clock while  $eclk$  and  $\overline{eclk}$  represent the timing clock for 2Gb/s. From the figure, one can see that for 1Gb/s,  $eclk$  is right at the center of the data eye while  $\overline{eclk}$  is right on the data transition edge for 1Gb/s data waveform. In other words, 1Gb/s is operated on the SDR mode while 2Gb/s is on the DDR mode. From this observation, we can see that by properly directing the output of the deserializer to the phase control logic, the link can be operated at both 2Gb/s and 1Gb/s while using the same clocking circuit and the same clock rate.

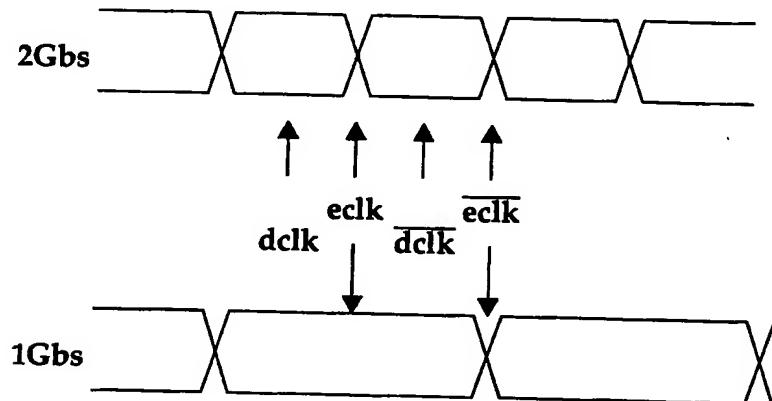
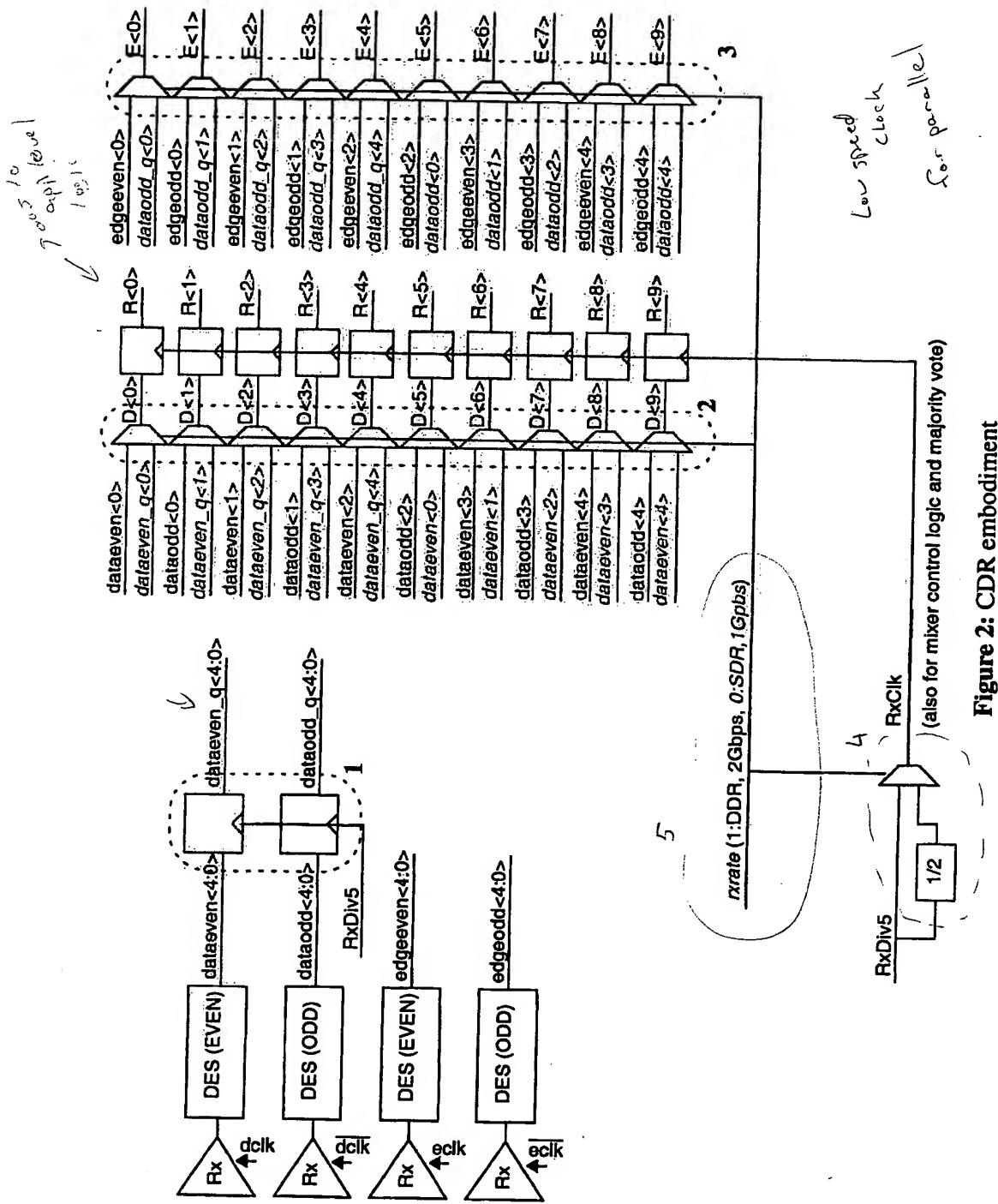


Figure 1: CDR Clocking for both 2Gb/s and 1Gb/s

Figure 2 shows such an embodiment. Four receivers clocked by  $dclk$ ,  $\overline{dclk}$ ,  $eclk$ ,  $\overline{eclk}$ , respectively.  $dataeven<4:0>$ ,  $dataodd<4:0>$  represent the deserializer output from the data receivers while  $edgeeven<4:0>$ ,  $edgeodd<4:0>$  represent the deserializer output from the edge receivers. The 10-bit data outputs are further registered by RxDiv5 (divide-by-5 clock), as shown in 1 in the figure. 2 and 3 are multiplexors added to select the data based on the high-speed (2Gb/s) or the low-speed (1Gb/s) operations.



**Figure 2:** CDR embodiment

clock cycle =  $\frac{1}{f_{clock}}$

Load pulse =  $\frac{1}{f_{clock}} + \frac{1}{f_{clock}} \times \frac{1}{2}$

Setup time =  $\frac{1}{f_{clock}} \times \frac{1}{2}$

Hold time =  $\frac{1}{f_{clock}} \times \frac{1}{2}$

Setup time + Hold time =  $\frac{1}{f_{clock}} + \frac{1}{f_{clock}} \times \frac{1}{2} + \frac{1}{f_{clock}} \times \frac{1}{2} = \frac{1}{f_{clock}} + \frac{1}{f_{clock}}$

$f_{clock} = \frac{1}{T}$

4

Notice that 1, 2, and 3 are circuits added for the dual-mode (DDR and SDR) operation. All other circuits remain without change for the DDR mode, including the phase mixer control logic not shown in the figure.

Referring back to Figure 2, the normal mode, i.e., DDR mode, is in normal characters while the SDR mode is in *italic* characters. In the normal DDR mode, the 10-bit parallel data  $D<9:0>$  are from dataeven and dataodd while the timing data  $E<9:0>$  are from edgeeven and edgeodd. In the SDR mode, only dataeven and dataodd are used. Please note that either data or edge can be used here. Ultimately the phase control loop will line up either the data or edge clocks to the center/edge of the data eye depending on which one is used. Furthermore, to be compatible with the 10-bit operation in the DDR mode, both dataeven and dataodd are registered by one divide-by-5 cycle to generate dataeven\_q and dataodd\_q. Dataeven and dataeven\_q are then directed to the D path while Dataodd and dataodd\_q are to the E path. The reason for the even path to D and the odd path to E is because in the original phase control loop, the data is received before the edge. In addition, the deserializer arranges the even data before the odd data because the original design use LSB first. Nonetheless, all these legacy concern does not limit the design to be only the one shown in the figure. One should rearrange the data order or flow based on the system requirement.

Rxrate is used to select between the DDR (2Gb/s) and the SDR (1Gb/s) mode. The clock for the parallel interface, RxClk, is divided by two in the SDR mode so that RxClk has the same frequency as  $R<9:0>$ .

The embodiment shows that there is no circuit added at the high-speed clock domain, i.e., dclk and eclk pairs. Hence, there is no added jitter for the CDR input. Since the circuits are added in the divide-by-5 clock domain, there should be minimum impact on the critical path due to the added multiplexors. There are only ten added flip-flops in 1 for dataeven\_q and dataodd\_q. This added loading on RxDiv5 should not be too significant, provided that RxDiv5 is responsible for driving the phase control logic so there are plenty of flip-flops already. From the physical design standpoint, the added 20 multiplexor and 10 flip-flops can be laid out in a very compact fashion between the deserializer output and the

phase control input. Due to the datapath nature of the circuit, the area impact should be minimum.

## 2.2 Transmitter Path Embodiment

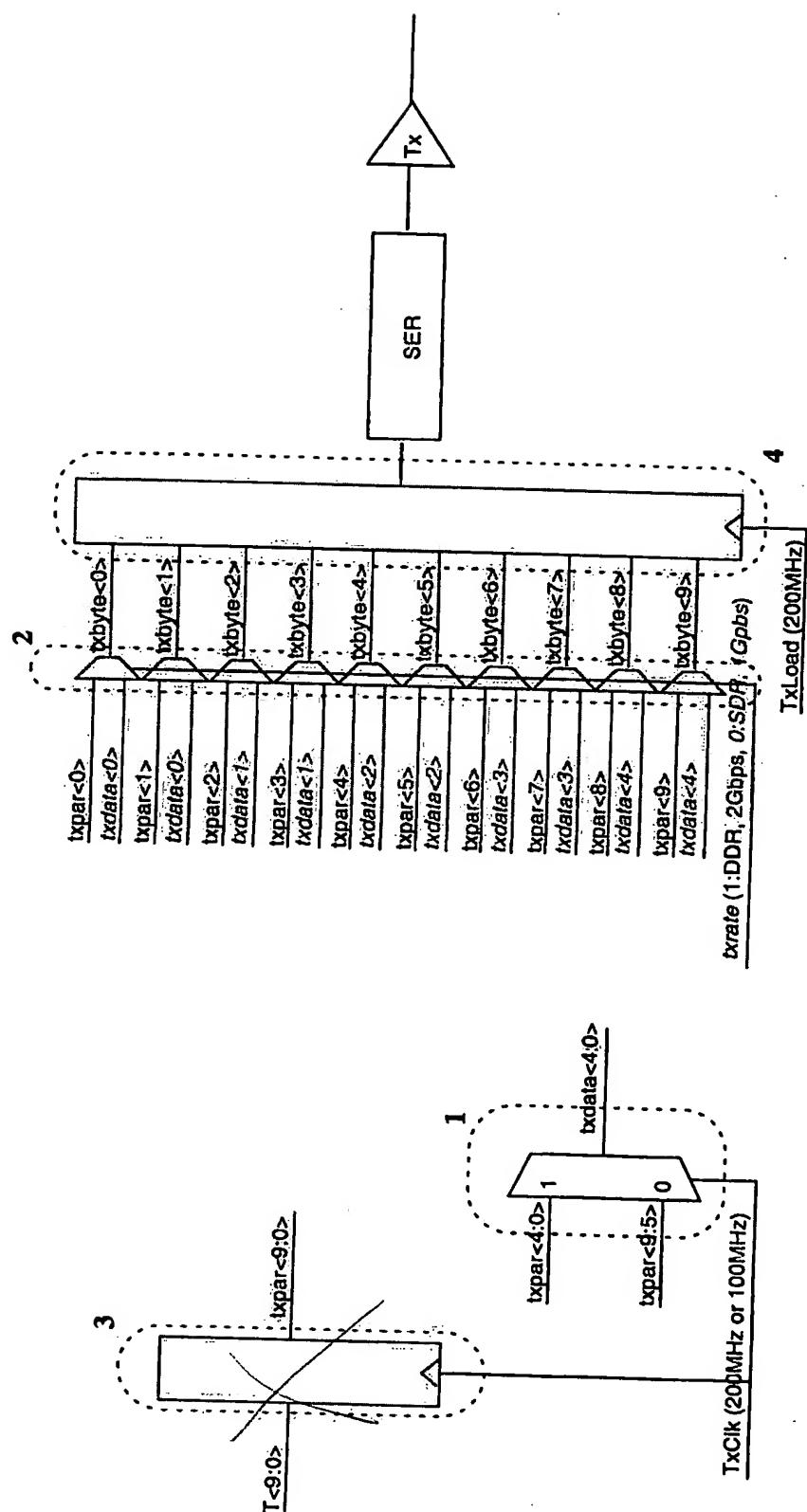
The design complexity on the transmit side is much simpler than the receiver/CDR side because unlike the CDR where the implementation may have impact on the phase control loop (and it does not in the embodiment), the transmit clock is directly from VCO, running at 1GHz for 2Gb/s and 1Gb/s operation.

Again, to simplify the discussion, the FiberChannel example is used. Therefore, as before, the VCO runs at 1GHz. And the requirement is to support both 2Gb/s and 1Gb/s on the same hardware.

Similar to the CDR embodiment, the transmitter path embodiment is focused on avoiding affecting the high-speed clock jitter. Therefore, the transmitter clock is not touched and always runs at 1GHz. To implement SDR from a DDR circuit, one can simply duplicate the bits for both even and odd.

Figure 3 shows an embodiment. Again, the DDR is in the normal characters while the SDR is in the *italic*. The parallel interface (3) is clocked by TxClk, which is at the same frequency as the parallel input,  $T<9:0>$ . The other parallel interface (4) is clocked by TxLoad, which is always at the DDR rate (200MHz in the example). TxLoad registers the data into the serializer, which always runs at 1GHz.

In the DDR mode,  $txpar<9:0>$  is directly sent to the multiplexors (2) to generate  $txbyte$ . In the SDR mode,  $txpar<9:0>$ , which runs at 100MHz in the example, is available for two TxLoad (200MHz) cycles. The multiplexors (1) first selects the lower 5 bits from  $txpar$  in the first half of TxClk cycle and then selects the higher 5 bits from  $txpar$  in the second half of TxClk cycle. The output of the multiplexors (1),  $txdata$ , is duplicated in pair and sent to the multiplexors (2). As a result, the transmitter transmits the same back-to-back data in the SDR mode.



**Figure 3:** Transmitter Path Embodiment

As shown in Figure 3, the transmitter path embodiment does not add any circuit on the high-speed transmit clock domain. Therefore, its jitter is not affected. Similar to the CDR embodiment, all the circuits are running in the divide-by-5 domain and hence should have minimum impact on the critical path. Only 1 and 2 are added and are totally 20 multiplexors. These datapath-like circuit can be laid out compactly between the serializer and the 10-bit interface.

Finally, this embodiment is only an example of how one can implement based on the current system constraint, such as the clock frequency, clock divide ratio, parallel interface width, etc. One should be able to adjust the circuits based on other system constraint.

### 3. Applications

The invention does not only provide a solution for the dual-rate requirement, it also relaxes the wide range requirement for the PLL design. For the latter part, one can set the VCO to run at the highest rate necessary for the DDR mode. When the link runs at lower speed (1/2, 1/3 or even lower), one can switch to SDR mode. This reduces the bit rate by 1/2 without any change in the PLL design.

Without this invention, one may have to design two flavors of PLL due to the minimum saturation margin constraint at high-speed and minimum gate overdrive constraint at low speed.